Plug-in Electric Vehicle Behavior Modeling in Energy Market

A Novel Deep Learning-Based Approach with Clustering Technique

Jahangir, Hamidreza; Sadeghi Gougheri, Saleh ; Vatandoust, Behzad ; Aliakbar Golkar, Masoud; Ahmadian, Ali; Hajizadeh, Amin

Published in:
I E E E Transactions on Smart Grid

DOI (link to publication from Publisher):
10.1109/TSG.2020.2998072

Publication date:
2020

Document Version
Accepted author manuscript, peer reviewed version

Plug-in Electric Vehicle Behavior Modeling in Energy Market: A Novel Deep Learning-Based Approach with Clustering Technique

Hamidreza Jahangir, Saleh Sadeghi Gougheri, Behzad Vatandoust, Masoud Aliakbar Golkar, *Senior Member*, *IEEE*, Ali Ahmadian, and Amin Hajizadeh, *Senior Member*, *IEEE*

***Abstract*— Growing penetration of Plug-in Electric Vehicles (PEVs) in the transportation fleet and their subsequent charging demands introduce substantial intermittency to the electric load profile which imposes techno-economic challenges on power distribution networks. To address the uncertainty in demand, a novel deep learning-based approach equipped with a hybrid classification task is developed which can take into account the travel characteristics of the PEV owners. The classification structure helps us scrutinize the PEVs demand by allocating a specific forecasting network to each cluster of travel behavior patterns. In our hybrid classification task, first, an unsupervised classifier discerns hidden travel-behavior patterns between the historical PEVs data by clustering them; then, a supervised classifier directs each new PEV data to its appropriate cluster-specific forecasting network. The deep learning-based forecasting and classification networks are constructed based on the Long Short-Term Memory networks to investigate long- and short term features in PEV behaviors. The data-driven structure of our proposed method enables us to observe and preserve the correlation between PEV travel data parameters (departure time, arrival time and traveled distance) and avoid the generation of unrealistic travel samples found in scenario-based approaches. To verify the effectiveness of the proposed method in practical environments, we have studied the impact of the precise forecasting of the PEVs demand in an aggregator's financial profit in the energy market of the California Independent System Operator market. The numerical results confirm the outstanding performance of our proposed deep learning-based method in forecasting PEVs demand against benchmark approaches in this field such as Monte Carlo, Quasi-Monte Carlo, and Copula with only a 6.77% error in comparison with real data.**

***Index Terms*—Deep learning; Classification; Plug-in Electric Vehicles; Travel behavior; Energy market.**

NOMENCLATURE

Sets

$\mathcal{A}$	Set of sample vector of each cluster, indexed by a
$\mathcal{G}$	Set of output layer sample, indexed by g
$\mathcal{K}$	Set of Cluster, indexed by $k, \hat{k}$
$\mathcal{L}$	Set of hidden layers, indexed by l
$\mathcal{P}$	Set of Plug-in electric vehicles (PEVs), indexed by p
$\mathcal{S}$	Set of state, indexed by s
$\mathcal{T}$	Set of bidding time intervals (hour), indexed by t [1,24]
$\hat{\mathcal{T}}$	Set of fifteen-minute market time intervals, indexed by $\hat{t}$ [1,4]
A_p	Set of time intervals in which PEV p is connected to charger, $A_p \subset t, \hat{t}$
Ta_p	Arrival time of PEV p, $p \in \mathcal{P}$ and $Ta_p \subset t, \hat{t}$

Parameters

α_p	Minimum state of charge PEV p in departure time (%)
$\hat{\Delta}t$	Length of fifteen minutes time interval (hour)
η_p	Charger efficiency of PEV p charger
ρ_t^{DA}	Day-ahead energy market (DAM) price in hour t ($/kWh)
$\rho^{EENC,DA}/\rho^{EENC,RT}$	Expected energy not charged (EENC) penalty cost in day-ahead/real-time market ($/kWh)
$\rho^{PEN,RT}$	Penalty value of not honoring DAM bids in real-time energy market (RTM) ($/kWh)
$\rho_{t,\hat{t}}^{RT}$	RTM energy price in hour t and time interval $\hat{t}$ ($/kWh)
BC_p	Battery capacity of PEV p (kWh)
CR_p	Rated charger capacity of PEV p (kW)
$epoch_{max}$	Total number of training epochs
Kmax	Maximum number of clusters
N_k	The number of members in cluster k
n_o	Total number of outputs
nh	Total number of hidden layers
P_v	Patient factor of the validation task
r	Dimension of each input data sequence
SOC_p^{ini}	Initial state of charge (SOC) of PEV p (kWh)

Variables

b^{out}	Bias vector of output layer
bc^l, bf^l, bi^l, bo^l	Bias vector for cell block, forget, input, and output gates of layer l
C_k	Centroid of cluster k
${c_s}^l$, ${f_s}^l$, ${i_s}^l$	Data vector of cell block, forget and input gates of layer l at state s
$Cost^{CH,DA}$	Day-ahead charging cost ($)
$Cost^{DA}/Cost^{RT}$	Total day-ahead/real-time aggregator cost ($)
$Cost^{EENC,DA}/$ $Cost^{EENC,RT}$	Day-ahead/real-time EENC cost ($)
$Cost^{INC,RT}$	Cost of placing incremental bids in RTM ($)
$Cost^{PEN,RT}$	Penalty cost of not honoring DAM bids in RTM ($)
d_{kk}	Dispersions of the cluster k
$d_{k\hat{k}}$	Distance between mean value of the cluster k and $\hat{k}$
$D_{k\hat{k}}$	Within-to-between distance ratio of cluster k and $\hat{k}$
$K_{optimal}$	Optimal number of clusters
$Loss_s^{epoch}$	Training loss of each epoch at state s
${o_s}^l$, ${o_s}^{nh}$	Data vector of output gate at state s of layer l and last layer
Out_s	Output vector at state s
$P_{t,\hat{t}}^{INC,RT}$	Incremental energy bids in RTM in hour t and time interval $\hat{t}$ (kW)
$P_{t,\hat{t}}^{PEN,RT}$	Amount of DAM awarded bid not consumed in RTM in hour t and time interval $\hat{t}$ (kW)
$pch_{p,t,\hat{t}}^{DA}/pch_{p,t,\hat{t}}^{RT}$	Day-ahead/real-time demand of PEV p in hour t and time interval $\hat{t}$ (kW)
PCH_t^{DA}	Day-ahead demand of PEV fleet in hour t (kW)
$PCH_{t,\hat{t}}^{RT}$	Real-time demand of PEV fleet in hour t and time interval $\hat{t}$ (kW)
S_s^l	State vector of layer l at state s
$SOC_{p,t,\hat{t}}^{DA}/SOC_{p,t,\hat{t}}^{RT}$	Day-ahead/real-time SOC of PEV p in hour t and time interval $\hat{t}$ (kWh)
$target_s$	Target vector at state s

Hamidreza Jahangir, Saleh Sadeghi Gougheri, Behzad Vatandoust, and Masoud Aliakbar Golkar are with the Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran. (Emails: h.r.jahangir@email.kntu.ac.ir, salehsadeghi@email.kntu.ac.ir, b.vatandoust@ee.kntu.ac.ir, golkar@kntu.ac.ir).

Ali Ahmadian is with the Department of Electrical Engineering, University of Bonab, Bonab, Iran (Email: ahmadian@bonabu.ac.ir).

Amin Hajizadeh is with the Department of Energy Technology, Aalborg University, Esbjerg, Denmark (Email: aha@et.aau.dk).

$Whi^l, Wh\varphi^l, Why^l, Who^l$	Weight vector for output of previous state input gate, forget gate, cell block, and output gate of layer l
$Wi^l, Wi\varphi^l, Wiy^l, Wio^l$	Weight vector for input of current state input gate, forget gate, cell block, and output gate of layer l
$Wout$	Weight vector of output layer
y_{ak}	Sample vector of cluster k
Y_g	Output vector of sample g
$\hat{Y}_g$	Desired vector of sample g
$\bar{Y}$	Mean value of desired vector
z_k	Mean vector of cluster k

I. Introduction

A. Background and motivations

Transportation electrification in the form of plug-in electric vehicles (PEVs) is growing in popularity and is considered a promising approach to alleviate the impact of carbon emission on global sustainability [1]. From the viewpoint of the power system, the uncontrolled charging patterns of PEV owners demonstrate that the electrical demand imposed by PEVs has a high tendency to concur with the peak load of traditional electrical consumption and introduces substantial uncertainty and sharp variations to the expected load profile. To address the undesired effects of uncontrolled PEV charging, an intermediary body called an aggregator is proposed in the literature. Through charging plans and contracts, the aggregators aim to deliver charging demand of their contracted PEVs considering technical and economic perspectives. Each aggregator needs to devise plans that are lucrative for PEV owners and at the same time ensure its profit margin in competition with other aggregators [2]. The cornerstone of the aggregators' decisions is the quality of its PEVs demand forecasts [3]. The real-world travel data indicate that, in comparison with the traditional residential demand, the expected electrical demand of PEVs exhibits higher levels of individuality which is reflected in the diversity of their travel patterns. As the fraction of PEVs grows in the transportation fleet, the aggregators will encounter a "big-data" problem whose analytical modeling can become a formidable task. Problems of this size require new tools that can autonomously identify patterns and behaviors in the large number of samples [4].

B. Literature survey

The previous researches on PEV travel behavior modeling fall into two main categories: scenario-based approaches [2], [5]–[13] data-driven approaches [14]–[19]. The majority of scenario-based approaches adopt the Monte Carlo (MC) simulation method, which relies on the generation of a large number of random samples from a search space defined by the probability distribution of travel parameters (i.e. PEV departure time, arrival time, and traveled distance). The authors in [2], [5]–[7] have employed the well-known probability distribution functions such as Normal and Gaussian to generate samples for each travel parameter. The authors in [8] have employed a joint probability distribution function to generate the departure time and arrival time of the PEVs. In [9], to investigate the dynamic effects of the PEVs demand and wind energy in the power system stability, the Quasi-Monte Carlo (QMC) method has been employed. In QMC, unlike the MC method where the samples are randomly selected, the low discrepancy sequences criterion is utilized to select the samples that can cover all the solution space in a homogeneous manner. Although the QMC is a modified version of the MC and presents a better sampling procedure than the MC [20], none of them are able to factor the correlation between the travel parameters in their models. The studies in [10], [11] improved MC-based models by incorporation of Copula functions to account for the dependencies between travel parameter data sets. However, there are two fundamental shortcomings in MC-based approaches: first, they impose high computational costs; second, in these methods, the generated PEV demands are meaningful only when considered in bulk. The individual travel samples can be impossible in reality, for instance, travels with short departure and arrival timespans that correspond to long traveled distances (e.g. 15-minute commute that corresponds to 50 miles traveled distance). The authors in [12] and [13] developed a hybrid MC and Markov model to predict PEV travel behaviors. The Markov-chain method simulates travels by predefined states and possible transitions between these states which are defined based on expert-decision and experience. Moreover, this method is unable to observe the correlation between travel parameters; and, similar to MC methods, the Markov-chain method also relies on voluminous sample generation to cover the problem search space.

In the data-driven approaches, some studies have investigated the PEVs demand forecasting task using time series forecasting tools such as the autoregressive integrated moving average (ARIMA) models. For instance, in [14], the PEVs' historical travel behavior data were used to find their consequent demand, afterward, an ARIMA model was fitted to the PEVs demand profiles to forecast their future evolution. In the same manner, the authors in [15], have utilized the fractional ARIMA (fARIMA) to investigate the seasonal trends in the PEVs demand. In their approach, the daily driving patterns of the PEV owners were only analyzed in the process of calculating the historical demand fed to the fARIMA model, and they were not directly reflected in the forecasting stage. This approach in PEVs demand forecast entails large uncertainty in the possible behavior of the PEV fleet and the temporal characteristics of individual PEV demand—which would ultimately render such approaches ineffective solutions in high PEVs penetration scenarios. Recently, a limited number of studies have utilized data-driven approaches based on neural networks (NNs) to overcome the aforementioned deficiencies in PEVs travel behavior forecasting [16]–[19]. In [16], a hybrid NN-based approach with limited layers based on nonlinear autoregressive model with exogenous inputs (NARXI) has been presented. In this study, the general travel behavior of the PEVs according to the road data has been modeled, however, the proposed method employed shallow NN which has low performance in feature extraction task and PEVs effect on the distribution power system is not considered. In this way, in [17] the authors employed a MC-based sample generation method to train their shallow NN. However, using the samples generated by MC would undermine the accuracy of NN training and, consequently, the final results. To improve the accuracy of training, the study in [18] used real-world driving data to train their shallow rough NN which, in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.2998072, IEEE Transactions on Smart Grid

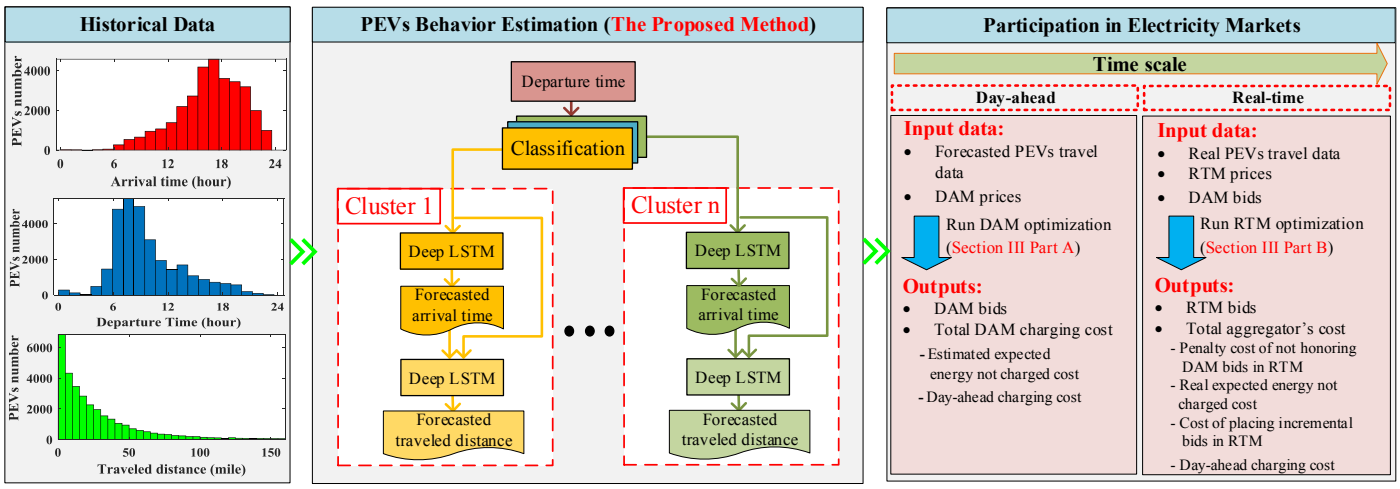

Fig. 1. Overall structure of the proposed method and energy market framework

comparison to basic NN, is able to handle the noise in input travel data more efficiently. Authors in [19] have employed a data-driven approach to cluster the PEVs travel data constructed based on the correlation between different travel behaviors. Their approach only divided travels based on the statistical parameters and did not predict the travel behavior of the PEV owners. The mentioned studies have employed shallow NNs, which by the structure are incapable of extracting the main features of large data sets such as PEV travel data [21]. Also, they did not investigate the possible hidden travel patterns in their PEV electric demand calculations.

C. Paper contributions

In this paper, we introduce an artificial intelligence solution founded on deep-learning concepts to fill the mentioned deficiencies in PEV travel behavior forecasting. The deep learning has proven to be a powerful tool in large-dimension problems with complex interrelations such as video and image pattern recognition, audio processing [4], time-series forecasting [22], classification tasks [23], etc. The key factor that distinguishes the deep learning concept from the rest of the data-driven approaches lies in its outstanding ability to autonomously extract the main features of a large dimension phenomenon entirely from its historical data [24]. In essence, in our approach, we feed real-world PEV data to a deep classifier to autonomously cluster the data based on the hidden travel behavior patterns existing among them. Then, a deep network is allocated to each cluster to capture and forecast the unique behavior of each cluster. Moreover, to further improve the accuracy of the forecasting results, we utilize a deep Long Short-Term Memory (LSTM) network which is able to model both short-term variations in travel behavior as well as long-term trends in characteristics of PEV travel patterns thanks to its various operation gates [23]. The detailed steps of our proposed approach are outlined as follows (Fig.1):

- In the initial step, the travel data samples are clustered based on their corresponding departure time. The departure time is selected as the classification criterion because it is plausible to assume it will be readily available to the aggregators under the premise that the PEV owners declare their desired departure time to the aggregator in advance at the end of their last daily trip [25]. (However, concerns regarding PEV owners' privacy and its impact on their participation in aggregators scheduling plans [26], [27] lie out of the scope of this paper.)
- Then, a deep forecasting network is earmarked to each travel pattern cluster. This discretization helps us to capture the behavioral subtleties of each travel behavior pattern in a more efficient manner.
- In the final step, the arrival time in each cluster is forecasted based on its corresponding departure time, then the forecasted arrival time and the observed departure time are fed to another cluster-specific deep network to forecast their corresponding travel distance.

In this work, the core concepts of deep learning are exploited to devise a novel solution for precise PEV forecasting tailored to the needs of power system aggregators. Particularly, from the perspective of an aggregator aiming to enter electricity markets. Our proposed solution attempts to address the knowledge gap in the existing body of literature: over-simplistic data-driven models that cannot identify hidden travel behavior patterns [16]–[19] and lack of proper correlation modeling between travel parameters [5]–[7], [10]–[13]. To verify the robustness of our proposed approach, its performance against real data in terms of accuracy of the forecasting results, modeling the correlation between travel parameters, and the profit margin of aggregators' energy bids in a day-ahead market (DAM) of California Independent System Operator (CAISO) is compared with the performance of benchmark methods in this field such as MC, QMC, and Copula models. Indeed, our results illustrate the crucial role that forecast accuracy plays in aggregators' financial gains and ensuring their viability in the electricity market. The overview of this study including the proposed method and energy market framework is illustrated in Fig. 1. In this study, to find the global optimal solution, the optimal charging task is formulated as a mixed-integer linear programming (MILP) problem, which also incorporates a linear form of the AC load flow to enforce the voltage and current limits of our host distribution network [28].

D. Organization of the paper

The rest of the paper is organized as follows: Section II presents the proposed deep learning approach with details. Section III defines the PEV optimal charging procedure

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.2998072, IEEE Transactions on Smart Grid

according to the CAISO energy market rules. The case study definition and numerical results are described in Section IV. Section V concludes the finding of this paper.

II. Problem Statement

To manage the PEVs' optimal charging procedure, aggregators first need to estimate the day-ahead (DA) travel behavior of their contracted PEVs with a high degree of precision. Afterward, they need to optimize PEVs charging schedules considering techno-economic indices of the host power distribution network and the welfare of PEV owners—the latter two constitute the main responsibilities of the aggregators. Fig. 1 depicts the general layout of our proposed PEV forecasting solution which employs deep LSTM networks for classification and prediction tasks. This section first provides a general description of LSTM networks, and then delves into the details of our classification task, forecasting task, and the overall structure of our proposed method, respectively.

A. LSTM network

The Recurrent Neural Network (R-NN) is an improved form of NNs which exploits previous information of its input data by making connections (recurrent weights) between its output and hidden layers. However, the R-NNs have a gradient vanishing problem (a steep decline in the gradient norm for long-term components during training) which poses a serious obstacle in forecasting profiles with complex behavior such as PEVs travel patterns [22]. To overcome this problem, in this paper, we employ a more sophisticated version of the R-NNs known as an LSTM block (Fig. 2). Every LSTM block is equipped with three operation gates namely *input*, *output*, and *forget gates*. To construct a deep LSTM network, we need to stack several LSTM blocks in an order that the input data of each LSTM block at state *s* is the output of the same network at state *s-1* and the output of the preceding network at state *s*. The *input gate*'s responsibility is to remember the information of the new and previous steps, and the *forget gate* is designed to expurgate the trivial information from the memory unit, and the *output gate* is employed to elicit advantageous information from the memory unit [29]. In deep LSTM networks, hidden features will be propagated among different LSTM blocks in the training task; this learning aptitude makes deep LSTM networks a promising tool for learning the behavior of complex phenomena with high precision. The general equations of an LSTM block at layer l are presented as follows [30]:

$$i_s{}^l = \sigma\left(Wi^l\, S_s^{(l-1)}\right) + Whi^l\, S_{(s-1)}^l + bi^l \quad (1)$$

$$f_s{}^l = \sigma\left(Wi\varphi^l\, S_s^{(l-1)}\right) + Wh\varphi^l\, S_{(s-1)}^l + bf^l \quad (2)$$

$$c_s{}^l = f_s{}^l\, c_{(s-1)}^l + i_s{}^l\, tanh\,(Wi\gamma^l\, S_s^{(l-1)} + Wh\gamma^l\, S_{(s-1)}^l + bc^l) \quad (3)$$

$$o_s{}^l = \sigma\,(Wio^l\, S_s^{(l-1)}) + Who^l\, S_{(s-1)}^l + bo^l \quad (4)$$

$$S_s{}^l = o_s{}^l\, tanh\,(c_s{}^l) \quad (5)$$

Here, the LSTM block variables are defined as Wi^l, $Wi\varphi^l$, $Wi\gamma^l \in \Re^{r\times nh}$, Whi^l, $Wh\varphi^l$, $Wh\gamma^l \in \Re^{nh\times nh}$, bi^l, bf^l, bc^l, $bo^l \in \Re^{1\times nh}$ which will be tuned in the training task. In the proposed method, every deep LSTM network is constructed by stacking the LSTM blocks; each LSTM block (Fig. 2) is defined as a layer and the training procedure is implemented in a holistic manner. In this study, deep LSTM networks are employed in both the classification and forecasting tasks; the overall structure of the networks in both tasks are the same as each other except for the configuration of their last layer. In the forecasting networks, Rectified Linear Unit (ReLU) is defined as the activation function of the last layer and Mean Squared Error (MSE) function is implemented to calculate the training error; whereas, in the classification procedure, the SoftMax activation function—turns numbers into probabilities that collectively sum to one—and Categorical Cross-Entropy (CCE) function are implemented as activation function of the last layer and error calculation function, respectively. Further information regarding the SoftMax and the CCE functions can be found in [31]. To improve the robustness and stability of the proposed method and avoid overfitting problems in the training procedure, we append the L2 regularization term in the loss function and apply the dropout technique with 0.001 and 0.5 rates, respectively. The L2 regularization term adjusts the variations of the weights during the training procedure and prevents sharp fluctuations that cause convergence problems in the training task [22], [32]. Dropout is deemed as a promising technique in the deep learning concept; by dropping a neuron in the training procedure, it is temporarily eliminated from the total network with all of its connections, in this process, the overall robustness of the network is improved [33]. The training procedure is controlled by the validation technique along with the maximum iteration number as shown in **Algorithm 1**. In the same vein, the values for maximum epochs, validation frequency, and validation patience rate are considered as 1000, 10, and 10, respectively. Also, the initial weights of deep LSTM networks are selected based on the Glorot method which offers significant improvements on the convergence of the deep-learning based networks [34]. Furthermore, we employ a piecewise learning rate with a 0.95 dropping factor to improve the learning procedure and avoid local optimal points; in this way, we can have an adaptive learning rate in proportion to the

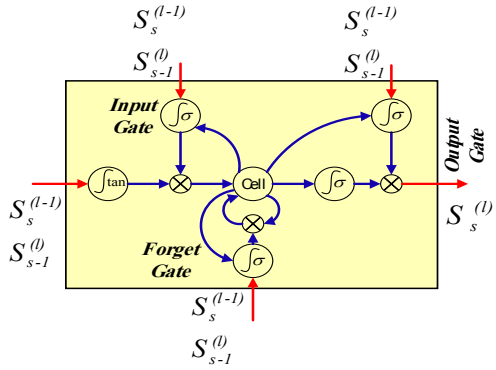

Fig. 2. The LSTM block with different operation gates

epoch number, which helps us reach a more stable training procedure.

B. Classification task

To discern the hidden travel patterns in the PEVs travel data, our approach relies on a hybrid classification task based on K-means (unsupervised) and deep LSTM network (supervised). As mentioned earlier, according to the existing communication capabilities of PEVs charging infrastructure, it is plausible to assume that the departure time of PEVs can be accessed by the aggregators. Therefore, we construct our classification task upon departure-time data of PEVs, which we envision to be adopted in practical scenarios. The overall procedure of the classification task is demonstrated in **Algorithm 2**. In the first step (**Algorithm 2: Part A**), we cluster departure-time data based on the K-means algorithm in an unsupervised manner for 1 to $Kmax$ number of clusters. The optimal number of clusters is then determined according to Davies-Bouldin (DB) index [35], which is defined as a ratio of within-cluster and between-cluster distances. In fact, DB index is calculated based on the worst-case separation for each cluster and averaging them as follows [35]:

$$D_{k\hat{k}} = \frac{d_{kk}+d_{\hat{k}\hat{k}}}{d_{k\hat{k}}} \tag{6}$$

$$d_{kk} = \left[\frac{1}{N_k}\sum_{a=1}^{N_k}\|y_{ak} - z_k\|^2\right]^{0.5} \tag{7}$$

$$d_{k\hat{k}} = \|z_k - z_{\hat{k}}\| \tag{8}$$

$$DB = \frac{1}{Kmax}\sum_{k=1}^{Kmax} Max_{k\neq\hat{k}}\, D_{k\hat{k}} \tag{9}$$

In the next step (**Algorithm 2: Part B**), after finding the optimal number of clusters using DB, the centroid of each cluster is used as its label for the supervised classification task in which we construct a deep LSTM classification network to match new departure-time data samples to their appropriate clusters.

C. Forecasting task

The main motivation behind classifying PEV data is to reduce the data size that each deep network is required to learn and forecast, which in turn increases the accuracy of the overall forecasts. In the previous step, we used a hybrid classifier to autonomously cluster PEVs travel data based on their departure time. In this step, to forecast the two remaining travel parameters i.e. arrival time and traveled distance, we allocate two deep LSTM networks to each cluster—one learns the mapping between departure time and arrival time to forecast arrival time

Algorithm 1
Training process of the Deep LSTM

Define: number of hidden layers (nh), maximum epoch number $epoch_{max}$, and patience factor for validation task (P_v)
Initialize: $Wi^l, Wi\varphi^l, Wi\gamma^l, Wio^l, Whi^l, Wh\varphi^l, Wh\gamma^l, Who^l, Wout, bi^l, bf^l, bc^l, bo^l, b^{out}$

1: **Begin operation** Back-Propagation
2: **Define:** k = 0
3: **while** Validation criterion is not satisfied (k<P_v) **do**
4: **for** *epoch* = 1 to $epoch_{max}$ **do**
5: **for** Each input sequence S_s in training data **do**
6: **for** l = 1 to nh **do**
7: Dropout neurons
8: Forward propagation: (Equations 1-5)
9: **end for**
10: Find output vector:
11: **if** Forecasting task **do**
12: $Out_s = ReLU(Wout\ o_s^{nh} + b^{out})$
13: **else if** Classification task **do**
14: $Out_s = softmax(Wout\ o_s^{nh} + b^{out})$
15: **end if**
16: **for** l = 1 to nh **do**
17: Update weights:
18: **if** Forecasting task **do**
19: $Loss_s^{epoch} = MSE(target_s, Out_s)$
20: **else if** Classification task **do**
21: $Loss_s^{epoch} = CCE(target_s, Out_s)$
22: **end if**
23: $Wi^l \leftarrow Wi^l - \nabla_{Wi^l}(Loss_s^{epoch})$
24: $Whi^l \leftarrow Whi^l - \nabla_{Whi^l}(Loss_s^{epoch})$
25: $Wi\varphi^l \leftarrow Wi\varphi^l - \nabla_{Wi\varphi^l}((Loss_s^{epoch})$
26: $Wh\varphi^l \leftarrow Wh\varphi^l - \nabla_{Wh\varphi^l}(Loss_s^{epoch})$
27: $Wi\gamma^l \leftarrow Wi\gamma^l - \nabla_{Wi\gamma^l}(Loss_s^{epoch})$
28: $Wh\gamma^l \leftarrow Wh\gamma^l - \nabla_{Wh\gamma^l}(Loss_s^{epoch})$
29: $Wio^l \leftarrow Wio^l - \nabla_{Wio^l}(Loss_s^{epoch})$
30: $Who^l \leftarrow Who^l - \nabla_{Who^l}(Loss_s^{epoch})$
31: $W_{out} \leftarrow W_{out} - \nabla_{W_{out}}(Loss_s^{epoch})$
32: Update biases same as weights
33: **end for**
34: **end for**
35: **for** Each input sequence S_s in validation data **do**
36: Run forward propagation: (line 8)
37: Calculate the validation loss:
38: **if** Forecasting task **do**
39: $Loss_s^{epoch} = MSE(target_s, Out_s)$
40: **else if** Classification task **do**
41: $Loss_s^{epoch} = CCE(target_s, Out_s)$
42: **end if**
43: **end for**
44: **If** $Loss_s^{epoch} > Loss_s^{epoch-1}$
45: $k = k + 1$
46: **else** $k = 0$
47: **end if**
48: **end for**
49: **end while**
50: **end operation**

Algorithm 2
Classification Task

Part A: Unsupervised Classification

1: **Begin operation**
Clustering travel data based on departure time (Unsupervised with K- means)
Input: Departure time data
Output: Optimal number of the clusters & centroid of each cluster
2: **Define** K_{max}
3: k=1;
4: **while** $k \leq K_{max}$ **do**
5: Unsupervised clustering the departure time with K-means algorithms by k Clusters
6: Calculate DB index
7: $k=k+1$;
8: **end while**
9: Find the optimal number of the clusters with the lowest DB index ($K_{optimal}$)
10: Designate centroid of each cluster as its label (C_k).
11: **End operation**

Part B: Supervised Classification

12: **Begin operation**
13: Classify each travel based on its departure time data and allocate a specific forecasting network to each cluster (Supervised with deep LSTM network)
Input: Departure time data
Target: Centroid of clusters
14: Match every input departure time data sample to its cluster with deep classification LSTM network based on centroid of each cluster as target
15: **End operation**

Algorithm 3
Forecasting Task

Part A

1: **Begin operation**
  Forecasting arrival time data for each cluster
  Input: Departure time data
  Output: Arrival time data
2: k=1;
3: **while** $k \leq \mathrm{K}_{\text{optimal}}$ **do**
4: Forecast the arrival time for *k-th* cluster with allocated LSTM network based on departure time data
5: k=k+1;
6: **end while**
7: **End operation**

Part B

8: **Begin operation**
  Forecasting traveled distance data for each cluster
  Input: Departure time data & arrival time data
  Output: Traveled distance data
9: k=1;
10: **while** $k \leq \mathrm{K}_{\text{optimal}}$ **do**
11: Forecast the traveled distance data for *k-th* cluster with allocated LSTM network based on departure and arrival time data
12: k=k+1;
13: **end while**
14: **End operation**

(**Algorithm 3: Part A**), and the other one learns the mapping between departure time, arrival time, and traveled distance to forecast traveled distance (**Algorithm 3: Part B**).

In the test procedure, the two deep LSTM networks perform in the following sequence: the first deep LSTM network takes as input the departure-time data provided by the deep classifier and forecasts the corresponding arrival times. Then, the forecasted arrival times in conjunction with corresponding departure times are fed to the second deep LSTM network to forecast the traveled distance pertaining to each data pair. In this way, the correlation between departure time, arrival time and traveled distance is preserved in the forecasted travel parameters, thus the possibility of forecasting unrealistic travels is eliminated. Observing the correlation between travel parameters greatly affects the accuracy of forecasts which is demonstrated and verified in Section IV.

III. PEVs Charging in Energy Market Framework

The main goal of the optimal charging procedure is to minimize the charging cost of the PEV owners in order to facilitate both the integration of PEVs into the power system and their adoption. In our work, it is assumed that the aggregator tries to minimize the PEVs charging costs by utilizing their demand flexibility and procuring their expected charging demand in DAM and real-time market (RTM)—here, we adopt the market rules of the CAISO to regularize market interactions. In the CAISO energy market, the awarded DAM bids are binding and the market participants are required to consume their awarded bids in the following day. Nevertheless, the participants may opt to adjust their submitted DAM bids in the RTM, however, they can only submit incremental bids. If the participants fail to honor their DAM awards they would incur payment recession and penalty costs [28]. Within this market context, the aggregator first needs to estimate its DA energy demand and submit DAM energy bids accordingly. Therefore, the accuracy of demand forecasts, in particular, PEVs' demand in our study, plays a crucial role in the economic gains of the aggregator in the energy market. Furthermore, to include the welfare of the PEV owners in the aggregators' objective function, we formulate the PEVs expected energy not charged (EENC) as a penalty cost for the aggregator.

A. DAM optimization

The aggregator solves the optimization problem (10) to determine its DAM energy bids, subjected to the constraints (10e)–(10j):

$$\underset{pch^{DA}_{p,t,\hat{t}}}{minimize} \quad Cost^{DA} \tag{10}$$

$$Cost^{DA} = Cost^{CH,DA} + Cost^{EENC,DA} \tag{10a}$$

$$Cost^{CH,DA} = \sum_t PCH^{DA}_t \rho^{\mathrm{DA}}_{\mathrm{t}} \tag{10b}$$

$$PCH^{DA}_t = \sum_p \sum_{\hat{t}} pch^{DA}_{p,t,\hat{t}}\, \widehat{\Delta \mathrm{t}} \tag{10c}$$

$$Cost^{EENC,DA} = \sum_p (\mathrm{BC_p} - \mathrm{SOC}^{\mathrm{ini}}_{\mathrm{p}} - \sum_t \sum_{\hat{t}} pch^{DA}_{p,t,\hat{t}}\, \eta_{\mathrm{p}} \widehat{\Delta \mathrm{t}}) \rho^{\mathrm{EENC,DA}} \tag{10d}$$

$$0 \leq pch^{DA}_{p,t,\hat{t}} \leq \mathrm{CR_p} \qquad t, \hat{t} \in A_p \tag{10e}$$

$$SOC^{DA}_{p,t,\hat{t}} = SOC^{DA}_{p,t,\hat{t}-1} + \left(pch^{DA}_{p,t,\hat{t}}\, \eta_{\mathrm{p}}\right) \widehat{\Delta \mathrm{t}} \tag{10f}$$

$$SOC^{DA}_{p,t,0} = SOC^{DA}_{p,t-1,4} \qquad \forall\, t > Ta_p \tag{10g}$$

$$SOC^{DA}_{p,t,0} = \mathrm{SOC}^{\mathrm{ini}}_{\mathrm{p}} \quad \forall\, t = Ta_p \tag{10h}$$

$$SOC^{DA}_{p,t,\hat{t}} \leq \mathrm{BC_p} \tag{10i}$$

$$\mathrm{SOC}^{\mathrm{ini}}_{\mathrm{p}} + \sum_t \sum_{\hat{t}} pch^{DA}_{p,t,\hat{t}}\, \eta_{\mathrm{p}} \widehat{\Delta \mathrm{t}} \geq \alpha_{\mathrm{p}}\, \mathrm{BC_p} \tag{10j}$$

The objective function (10) aims to minimize the total aggregator DAM charging cost (10a), which consists of the DAM energy cost for energy bids submitted to meet PEVs charging demand (10b) and the EENC cost which is defined as a penalty cost that the aggregator is obliged to reimburse PEV owners for its inability to fully charge their PEV batteries as calculated by (10d). Equations (10e)–(10j) show the PEVs technical charging constraints. The maximum PEVs charging rate in each time step is enforced by (10e). The state of charge (SOC) of PEVs calculated based on (10f)–(10h), and its maximum value is limited by (10i). The welfare of PEV owners is observed (10j) in which the aggregator is required to bring the SOC of PEVs at departure to at least α_{p}% of their battery capacity.

B. RTM optimization

In the second part of our formulation, we have introduced another optimization problem (11), inspired by the CAISO RTM rules, to evaluate the performance of the aggregator on the day of bid deployment where it is exposed to real PEV travel parameters and has to make adjustments to its energy bids in order to meet PEVs demands while honoring its awarded DAM bids. The main goal of the problem (11) is to form a judgment about the overall performance of the aggregator in the energy market and the role that PEV demand forecasts play in thereof.

$$\underset{P^{INC,RT}_{t,\hat{t}}, P^{PEN,RT}_{t,\hat{t}},\, pch^{RT}_{p,t,\hat{t}}}{minimize} \quad Cost^{RT} \tag{11}$$

$$Cost^{RT} = Cost^{INC,RT} + Cost^{PEN,RT} + Cost^{EENC,RT} \tag{11a}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.2998072, IEEE Transactions on Smart Grid

$$Cost^{INC,RT} = \sum_t \sum_{\hat{t}} P_{t,\hat{t}}^{INC,RT} \, \widehat{\Delta t} \, \rho_{t,\hat{t}}^{RT} \quad (11b)$$

$$Cost^{PEN,RT} = \sum_t \sum_{\hat{t}} P_{t,\hat{t}}^{PEN,RT} \, \widehat{\Delta t} \, \rho^{PEN,RT} \quad (11c)$$

$$Cost^{EENC,RT} = \sum_p (BC_p - SOC_p^{ini} - \sum_t \sum_{\hat{t}} pch_{p,t,\hat{t}}^{RT} \, \eta_p \widehat{\Delta t}) \rho^{EENC,RT} \quad (11d)$$

$$PCH_t^{DA} - P_{t,\hat{t}}^{PEN,RT} + P_{t,\hat{t}}^{INC,RT} = PCH_{t,\hat{t}}^{RT} \quad (11e)$$

$$0 \le P_{t,\hat{t}}^{INC,RT}, 0 \le P_{t,\hat{t}}^{PEN,RT} \le PCH_t^{DA} \quad (11f)$$

The objective function (11) aims to minimize the aggregators' costs in RTM, $Cost^{RT}$, which itself consists of three costs (11a): cost of procuring additional energy demand from RTM, $Cost^{INC,RT}$ (11b); the cost of not being able to consume the awarded DAM bids, $Cost^{PEN,RT}$(11c); and, the EENC for the PEVs on the day of bid deployment, $Cost^{EENC,RT}$(11d). Equation (11e) reflects the biding nature of the awarded DAM bids. Also, the problem (11) includes the same constraints expressed in equations (10e)–(10j), which we omitted for the sake of brevity—please note that in the counterpart equations of the problem (11) all the *DA* superscripts are substituted with real-time(*RT)* superscripts. The total aggregator's cost is calculated by the sum of the DAM (10a) and RTM (11a) costs. Lastly, we incorporate a linear form of AC load flow in both problems (10) and (11) to observe the voltage and current limit of our host distribution network. We strongly encourage our reader to see reference [28] for detailed formulation.

IV. Numerical Results

A. Data description

For our numerical simulations, we used 10000 travel data of PEVs obtained from the 2017 National Household Travel Survey (NHTS) [36] to train our proposed deep learning-based approach (80% for training, 10% for validation, and 10% for test); it should be noted that the data segmentation is done according to the vehicle samples. After the training task, to investigate the robustness of the proposed method, a case study with 1000 PEVs is considered. To develop the proposed optimal charging task a medium voltage distribution network with 21 buses is considered whose data and topology are presented in [28]. Moreover, to model our inflexible residential load in the network, we obtained a real load profile within CAISO market. Similarly, the DA and RT energy prices are also obtained from CAISO, for October 30, 2018 [37]. Also, the penalty price for consumption lower than the awarded DA bids, $\rho^{PEN,RT}$, is set to 30% of the maximum energy price of the given day. In addition, to incorporate the welfare of the PEV owners in our objective function we set the minimum PEV departure SOC to 75%. Also, the penalties for the amount of EENC to PEVs in DA and RT markets (i.e. $\rho^{EENC,DA}$ and $\rho^{EENC,RT}$) are set to 50% of the maximum energy price of their respective markets. The value of $\widehat{\Delta t}$ is defined as $\frac{1}{|\hat{\gamma}|}$. Finally, to match the granularity of PEV travel parameters to the bidding interval of the RTM of CAISO, we model PEVs arrival and departure time in 15-minute intervals. CPLEX (version 12.9) in GAMS optimization software was used to solve the MILP market optimization problems.

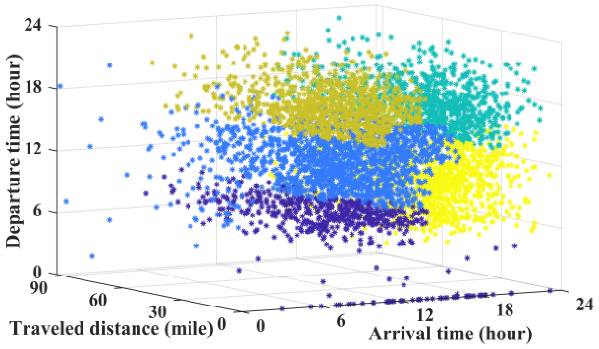

Fig. 3. The PEVs travel behavior classification results

B. Training framework

Classification Learner and Deep Network Designer toolboxes of MATLAB 9.6 were used to perform the classification tasks and the training procedure of the LSTM networks, respectively [38]. Our deep LSTM networks in classification and forecasting tasks are constructed based on 100 and 300 LSTM blocks (nh), respectively, which are stacked together to build a deep LSTM network, and $Kmax$ is set as 15.

C. Evaluation criterion

To evaluate the accuracy of the proposed method against real data we used the R-squared criterion which is defined as the square of the correlation between the target and the forecasted values. The R-squared value is between [0,1] in which higher values imply higher forecasting quality [22].

$$R^2 = 1 - \frac{\sum_{g=1}^{n_0} (Y_g - \hat{Y}_g)^2}{\sum_{g=1}^{n_0} (Y_g - \bar{Y})^2} \quad (12)$$

D. Simulation results

In this part, we want to illustrate the two main features of our proposed deep-learning based method which distinguish it from the existing literature: autonomous travel pattern identification (travel data classification) and accounting for correlation between travel data parameters (observing the realistic relationship between traveled distance, arrival time, and departure time in forecasting task). Therefore, in the first part of simulation results, we demonstrate the accuracy of the classification task and then draw a comparison between our proposed method with and without classification task (hereafter, the classified and unclassified cases are called C-Deep and UC-Deep, respectively), in terms of R-squared indices of the forecasted arrival time and traveled distance. Afterward, in the second part of simulation results, we discuss the effectiveness of

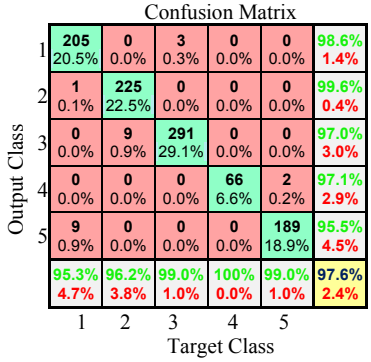

Fig. 4. The numerical results of the PEVs travel behavior classification

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.2998072, IEEE Transactions on Smart Grid

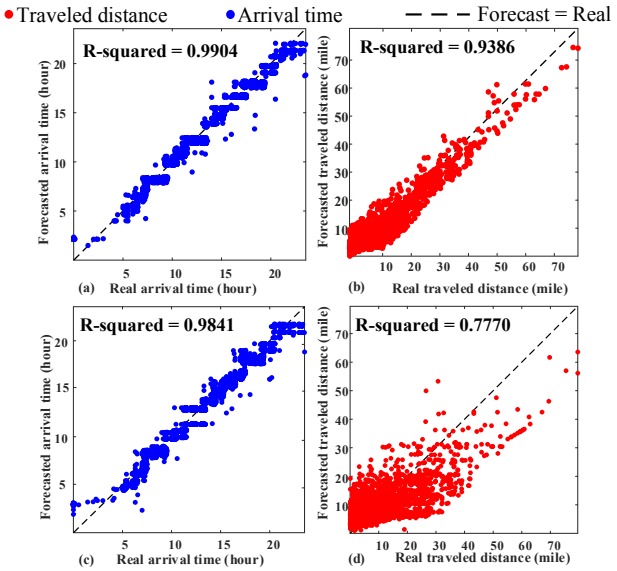

Fig. 5. Regression plots of forecasted PEV travel parameters in classified case (C-Deep) and in unclassified case (UC-Deep): (a) arrival time of C-Deep, (b) traveled distance of C-Deep, (c) arrival time of UC-Deep, and (d) traveled distance of UC-Deep

accounting for correlation in forecasting task by drawing a comparison between the performance of our proposed method and other benchmark methods in the literature. The overall classification results are illustrated in Fig. 3. As shown in Fig. 3, by clustering the PEVs travel data based on the departure time, we have five different clusters which present five general hidden travel patterns in our data set. It should be mentioned that the optimal number of clusters and centroids of each cluster are determined based on DB index and K-means algorithms, respectively, and this procedure is done autonomously with the unsupervised classification task according to the hidden pattern of the input data (**Algorithm 2 Part A**). To evaluate the supervised classification task, which is done by deep LSTM network, the confusion matrix of the classification result is presented in Fig. 4. As shown in Fig. 4, the classification precision in each cluster is between 95.3% to 100%, and the overall accuracy of the supervised classification task is 97.6% which verifies the robustness of the classification procedure (**Algorithm 2 Part B**). After the classification task, a specific deep LSTM network is allocated for each cluster to forecast arrival time (**Algorithm 3 Part A**), and traveled distance (**Algorithm 3 Part B**), separately. To verify the effectiveness of the classification task we demonstrate the forecasted results of our proposed method C-Deep (Fig. 5(a)–(b)), and UC-Deep (Fig. 5(c)–(d))—in the UC-Deep we simply bypass (**Algorithm 3 Part A**) and directly feed the unclassified travel data to **Algorithm 3 Part B**. According to Fig. 5, the effect of classification on the forecast accuracy is more pronounced in the traveled distance than arrival time. While the forecast accuracy

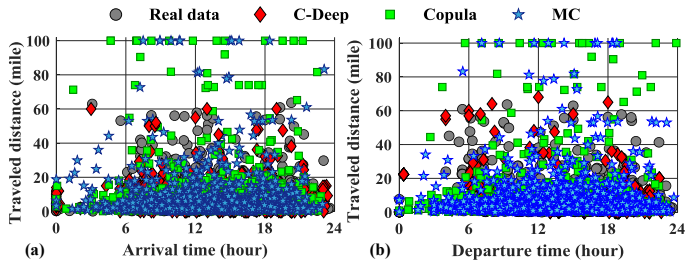

Fig. 6. PEVs travel behavior forecasting results based on different approaches

TABLE I. TOTAL CHARGING COST OF THE PEVS IN A DAY

Different costs	Methods					
	Real data	*C-Deep*	*UC-Deep*	*Copula*	*QMC*	*MC*
DA PEVs charging cost ($)	392.11	406.85	416.02	443.54	450.35	456.20
Penalty cost ($)	3.21	17.27	24.38	47.72	51.12	56.45
RT PEV charging cost ($)	23.11	23.11	23.60	19.76	16.94	16.72
RT EENC cost ($)	4.08	3.92	3.63	2.14	0.80	0.72
Total PEVs charging cost ($)	422.51	451.15	467.63	513.16	519.21	530.09
Accuracy of charging cost estimation (%)	------	93.23	89.33	78.55	77.11	74.54

of arrival time is acceptable in both cases (R-squared value of 0.99 for C-Deep and 0.98 for UC-Deep), there is a substantial improvement in the forecast accuracy of the traveled distance (R-squared value of 0.93 in C-Deep and 0.77 in UC-Deep). In fact, with classification task (C-Deep), for each travel-pattern cluster, a specific forecasting network is allocated; in this way, each forecasting network is only responsible for a limited area of the problem space and is trained in a more exclusive way in comparison with UC-Deep that a single forecasting network is responsible for the entire problem space. The significance of this improvement in the performance of the aggregator is further scrutinized at the end of this section. To verify the performance of our proposed method in accounting for the correlation between travel parameters, we compare our forecasted results against the real travel data and benchmark methods including MC, QMC, and Copula. Fig. 6 summarizes the performance of all four methods in the bulk generation of PEVs travels data against real travel data. The key difference between the generated samples of our proposed method and that of the benchmark methods is its ability to generate realistic travel patterns. In Fig. 6, the travel samples that fall into the rear ends of arrival time, departure time, and traveled distance axes are either infeasible or unrealistic. In contrast, Fig. 6 shows that our proposed method follows the trend of real-world travel data effectively. To investigate the financial ramification of the forecast errors for the aggregators, we have evaluated the performance of different approaches in real market conditions. In the real market, the aggregators first need to submit their DA energy bids, which they must honor in the following day otherwise they would incur penalties for not doing so. Fig. 7 depicts the DA energy bids (kW) calculated by all five cases (i.e. C-Deep, UC-Deep, Copula, QMC, and MC), and DAM price ($/kWh). The results demonstrate that our proposed method (C-Deep) outperforms other benchmark methods in predicting the PEVs DA load demand. Fig. 7 shows the intuitive behavior that in all cases the aggregator aims to meet its PEV demand in the cheapest hours, however, the realization of thereof requires accurate demand predictions. In order to evaluate the consequences of demand prediction errors in practical scenarios, we have calculated the penalties (for lower consumption than awarded DAM bids) that each method would incur in the day of bid deployment when they deal with the real PEV demand (Fig. 8(a) – Fig. 8(e), and Table I. Ideally, it is desirable for the aggregator to not deviate from its awarded DA bids. Comparing the performance of C-Deep and UC-Deep (Fig.8(a) and Fig. 8(b), respectively) in terms of RTM costs (i.e. underconsumption penalty and incremental RT adjustment bids) further demonstrates the effectiveness of classification. In comparison to UC-Deep, C-Deep results in 3.9% less overall aggregator cost;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.2998072, IEEE Transactions on Smart Grid

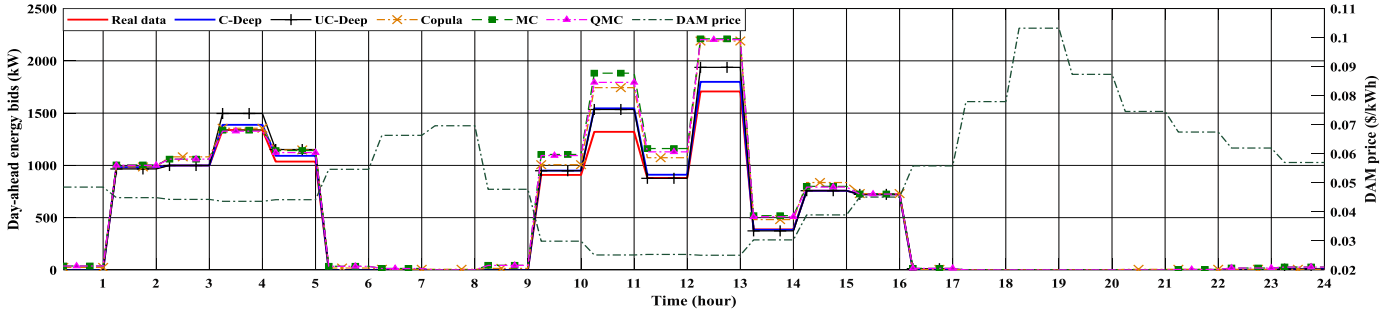

Fig. 7. Day-ahead demand bids of different cases (kW) with DAM price ($/kWh)

this improvement is derived from less overestimation in DA bids, which yields lower underconsumption penalty cost. Drawing a comparison between the performance of our proposed method with Copula, QMC and MC methods underlines the effectiveness of considering correlation in prediction accuracy. The Copula, QMC and MC methods greatly overestimated the DA PEV demand, so they incurred heavier penalties in contrast to C-Deep, and UC-Deep for not being able to consume the bids they submitted to the DA market—Fig. 8(c), Fig. 8(d), and Fig. 8(e), respectively. On the other hand, since our proposed method took into account the correlation between travel parameters and did not generate unrealistic travels, it is able to submit more realistic bids, which only incurs a minor penalty in comparison to the other benchmark methods. Table I, summarizes the performance of the investigated methods in DA and RT energy markets. The total PEVs charging cost which is the summation of DA energy cost, RT penalty, RT energy cost, and EENC cost proves the superiority of our proposed method. Table I also shows that the Copula method outperforms QMC and MC methods, which can be attributed to its ability to model statistical dependencies between travel parameter datasets; this outcome further underlines the importance of accounting correlation between travel parameters. As demonstrated in Table I, the QMC method, which is equipped with a modified sampling procedure, presents 77.11% accuracy in the charging cost estimation that verifies its superior performance against MC with 74.54 % accuracy. In summary, the ability of our proposed method to discern hidden travel patterns and utilize cluster-specific deep LSTM networks to estimate traveled distance based on its corresponding departure and arrival times and correlation of thereof enables it to ensure the feasibility of the estimated travel samples. The numerical results of this study underline the dependency of aggregator's financial profit margin on the accuracy of its PEVs demand prediction. If we assume a market where each of the mentioned methods was adopted by an aggregator it is clear that the aggregator employing our proposed deep learning approach would dominate the market with its superb performance and strong profit margin over the long term.

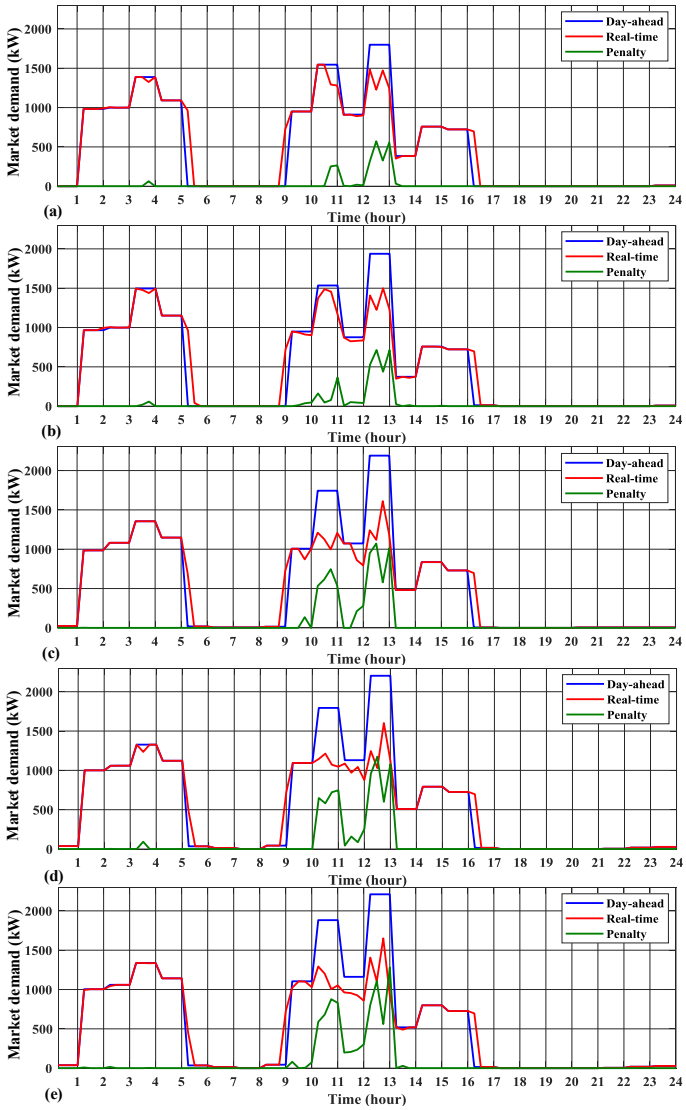

Fig. 8. Day-ahead bids, real-time bids, and penalty of different cases (kW): (a) classified deep-learning case (C-Deep), (b) unclassified deep-learning case (UC-Deep), (c) Copula case, and (d) Quasi-Monte Carlo case, (e) Monte Carlo case.

V. Conclusion

In this study, a novel deep learning-based approach with a hybrid classification task based on LSTM networks is presented to forecast the PEVs travel behavior and their electrical demand. The charging cost of the PEVs from the aggregator's perspective was investigated in DAM and RTM to verify the robustness of the proposed method in comparison with benchmark methods in this field i.e. Copula, QMC, and MC. In fact, the hybrid classification task was employed to autonomously discern the hidden travel patterns of the PEV owners (with 97.6% accuracy), so that an exclusive deep LSTM forecasting network could be allocated to each behavior cluster that resulted in $16.48 improvement of aggregator's daily cost employing the proposed method (C-Deep) in comparison with the unclassified case (UC-Deep). The numerical results also substantiated that our proposed approach with 93.23% accuracy in forecasting the PEVs' charging cost outperformed the scenario-based methods in our study i.e. Copula, QMC, and MC with 78.55%, 77.11%,

and 74.54% accuracy levels, respectively. The superior performance of the proposed method can mainly be attributed to its strong memory and feature extraction ability that enables it to capture the correlation between the different travel parameters (departure time, arrival time and traveled distance) and avoid the generation of unrealistic and infeasible travel samples. Our findings suggest that deep learning-based approaches offer great performance in PEV demand modeling and will substitute legacy scenario-based approaches in this filed. For future works, other energy markets such as ancillary services can be considered. Furthermore, by gathering person by person information of the PEV owners in a region over a long term, more details can be investigated in the clustering task such as weekdays, traffic condition, and so on.

REFERENCES

[1] T. Qian, C. Shao, X. Wang, and M. Shahidehpour, "Deep Reinforcement Learning for EV Charging Navigation by Coordinating Smart Grid and Intelligent Transportation System," *IEEE Trans. Smart Grid*, 2019.

[2] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-Free Real-Time EV Charging Scheduling Based on Deep Reinforcement Learning," *IEEE Trans. Smart Grid*, 2018.

[3] M. Carrión, "Determination of the selling price offered by electricity suppliers to electric vehicle users," *IEEE Trans. Smart Grid*, 2019.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[5] H. Fan, C. Duan, C.-K. Zhang, L. Jiang, C. Mao, and D. Wang, "ADMM-based multiperiod optimal power flow considering plug-in electric vehicles charging," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 3886–3897, 2017.

[6] J. Wang, G. R. Bharati, S. Paudyal, O. Ceylan, B. P. Bhattarai, and K. S. Myers, "Coordinated electric vehicle charging with reactive power support to distribution grids," *IEEE Trans. Ind. Informatics*, vol. 15, no. 1, pp. 54–63, 2018.

[7] J. Zhao, Z. Xu, J. Wang, C. Wang, and J. Li, "Robust distributed generation investment accommodating electric vehicle charging in a distribution network," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 4654–4666, 2018.

[8] O. Frendo, N. Gaertner, and H. Stuckenschmidt, "Real-Time Smart Charging Based on Precomputed Schedules," *IEEE Trans. Smart Grid*, 2019.

[9] H. Huang, C. Y. Chung, K. W. Chan, and H. Chen, "Quasi-monte carlo based probabilistic small signal stability analysis for power systems with plug-in electric vehicle and wind power integration," *IEEE Trans. Power Syst.*, 2013.

[10] B. Vatandoust, A. Ahmadian, and M. A. Golkar, "Stochastic copula-based multivariate modeling of plug-in hybrid electric vehicles load demand in residential distribution network," in *2016 Smart Grids Conference (SGC)*, 2016, pp. 1–7.

[11] E. Pashajavid and M. A. Golkar, "Optimal placement and sizing of plug in electric vehicles charging stations within distribution networks with high penetration of photovoltaic panels," *J. Renew. Sustain. Energy*, vol. 5, no. 5, p. 53126, 2013.

[12] Q. Yang, S. Sun, S. Deng, Q. Zhao, and M. Zhou, "Optimal Sizing of PEV Fast Charging Stations with Markovian Demand Characterization," *IEEE Trans. Smart Grid*, 2019.

[13] S. Sun, Q. Yang, and W. Yan, "A Novel Markov-Based Temporal-SoC Analysis for Characterizing PEV Charging Demand," *IEEE Trans. Ind. Informatics*, vol. 14, no. 1, pp. 156–166, Jan. 2018.

[14] M. H. Amini, A. Kargarian, and O. Karabasoglu, "ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation," *Electr. Power Syst. Res.*, 2016.

[15] N. Korolko, Z. Sahinoglu, and D. Nikovski, "Modeling and Forecasting Self-Similar Power Load Due to EV Fast Chargers," *IEEE Trans. Smart Grid*, 2016.

[16] K. Vatanparvar, S. Faezi, I. Burago, M. Levorato, and M. A. Al Faruque, "Extended range electric vehicle with driving behavior estimation in energy management," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2959–2968, 2018.

[17] D. Panahi, S. Deilami, M. A. S. Masoum, and S. M. Islam, "Forecasting plug-in electric vehicles load profile using artificial neural networks," in *2015 Australasian Universities Power Engineering Conference (AUPEC)*, 2015, pp. 1–6.

[18] H. Jahangir *et al.*, "Charging Demand of Plug-in Electric Vehicles: Forecasting Travel Behavior Based on a Novel Rough Artificial Neural Network Approach," *J. Clean. Prod.*, vol. 229, pp. 1029–1044, 2019.

[19] X. Li, Q. Zhang, Z. Peng, A. Wang, and W. Wang, "A data-driven two-level clustering model for driving pattern analysis of electric vehicles and a case study," *J. Clean. Prod.*, vol. 206, pp. 827–837, 2019.

[20] A. Singhee and R. A. Rutenbar, "Why quasi-Monte Carlo is better than Monte Carlo or Latin hypercube sampling for statistical circuit analysis," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, 2010.

[21] K. L. López, C. Gagné, and M.-A. Gardner, "Demand-side management using deep learning for smart charging of electric vehicles," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2683–2691, 2018.

[22] H. Jahangir *et al.*, "A Novel Electricity Price Forecasting Approach Based on Dimension Reduction Strategy and Rough Artificial Neural Networks," *IEEE Trans. Ind. Informatics*, pp. 1–1, 2019.

[23] S. Singh and A. Majumdar, "Non-intrusive Load Monitoring via Multi-label Sparse Representation based Classification," *IEEE Trans. Smart Grid*, 2019.

[24] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science (80-.).*, vol. 313, no. 5786, pp. 504–507, 2006.

[25] D. Federico, Y. Kondo, and D. T. Smith, "Departure time scheduling control system for an electric vehicle." Google Patents, 08-Sep-2015.

[26] M. H. Au, J. K. Liu, J. Fang, Z. L. Jiang, W. Susilo, and J. Zhou, "A new payment system for enhancing location privacy of electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 3–18, 2013.

[27] I. Bilogrevic, M. Jadliwala, K. Kalkan, J.-P. Hubaux, and I. Aad, "Privacy in mobile computing for location-sharing-based services," in *International Symposium on Privacy Enhancing Technologies Symposium*, 2011, pp. 77–96.

[28] B. Vatandoust, A. Ahmadian, M. A. Golkar, A. Elkamel, A. Almansoori, and M. Ghaljehei, "Risk-Averse Optimal Bidding of Electric Vehicles and Energy Storage Aggregator in Day-ahead Frequency Regulation Market," *IEEE Trans. Power Syst.*, 2018.

[29] J.-F. Toubeau, J. Bottieau, F. Vallée, and Z. De Grève, "Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1203–1215, 2018.

[30] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, 2017.

[31] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in neural information processing systems*, 2018, pp. 8778–8788.

[32] T. Van Laarhoven, "L2 regularization versus batch and weight normalization," *arXiv Prepr. arXiv1706.05350*, 2017.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, 2014.

[34] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[35] R. N. Khushaba, A. H. Al-Timemy, A. Al-Ani, and A. Al-Jumaily, "A Framework of Temporal-Spatial Descriptors-Based Feature Extraction for Improved Myoelectric Pattern Recognition," *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2017.

[36] "National Household Travel Survey," 2018. [Online]. Available: https://nhts.ornl.gov/. [Accessed: 05-Dec-2018].

[37] "California ISO Open Access Same-time Information System (OASIS)." [Online]. Available: http://oasis.caiso.com/mrioasis/logon.do. [Accessed: 17-Jan-2020].

[38] "D. L. Toolbox, Statictics, and M. L. Toolbox, version 9.6 (R2019a). Natick, Massachusetts: The MathWorks Inc., 2019." .